

```

;read_swath_l2.pro
;
; Reader for Level 2 Swath files
; Handles both Standard Product and Support Product.
;
; All data fields are not read by this example.
; Add keyword=parameter to install more data fields.
;
; A keyword cannot be a substring of another keyword.
; For example, Tsurf could not be used as a keyword if
; TsurfErr is also a keyword.
;
; Standard and the support files have some parameters
; in different unit, as well as different pressure grids.
;
;
pro read_swath_l2, pattern, numfp, numline, numpres, $
    tai, lat, lon, pres, tair, h2o, ozo, $
    Nsurf, Psurf, RetQAFlag, $
    TSurface=TSurf, $
    Tsurf_Air=TSurf_Air, $
    LandFrac=LandFrac, $
    sun_glint_distance=sun_glint_distance, $
    final_clear_flag=clear_flag, $
    invalid=invalid, $
    CldFrc=CldFrc, $
    PCldTop=PCldTop, $
    TCldTop=TCldTop, $
    clear_flag_4um=clear_flag_4um, $
    clear_flag_11um=clear_flag_11um, $
    tsurf_forecast=ts_forecast, $
    tsurf_diff_11um=tsurf_diff_11um, $
    spatial_coh_11um = spatial_coh_11um, $
    tsurf_diff_4um=tsurf_diff_4um, $
    spatial_coh_4um = spatial_coh_4um, $
    cldfrcvis=cldfrcvis

filename = findfile ( pattern, count = cnt )
if cnt ne 1 then begin
    print, 'findfile did not return exactly one file ', cnt, pattern
    if cnt eq 0 then return
endif

print, 'read_swath_l2: ', filename[0]

FAIL = -1
SWATH_L2_STD = 'L2_Standard_atmospheric&surface_product'
SWATH_L2_SUP = 'L2_Support_atmospheric&surface_product'

fid=EOS_SW_OPEN(filename[0], /READ)
if fid eq FAIL then return

SWid=EOS_SW_ATTACH(fid, SWATH_L2_STD)
if SWid eq FAIL then begin
    SWid=EOS_SW_ATTACH(fid, SWATH_L2_SUP)
    if SWid eq FAIL then return
    standard = 0
    print, "Reading "+SWATH_L2_SUP
endif else begin
    standard = 1
    print, "Reading "+SWATH_L2_STD
endelse

```

```

; read required fields which are common to both
; standard product and support product

ret = EOS_SW_READFIELD(SWid,"Time",tai,EDGE=edge)
sz=size(tai)
numfp=sz[1]
numline=sz[2]
ret = EOS_SW_READFIELD(SWid,"Latitude",lat)
ret = EOS_SW_READFIELD(SWid,"Longitude",lon)
ret = EOS_SW_READFIELD(SWid,"RetQAFlag", RetQAFlag)
ret = EOS_SW_READFIELD(SWid,"PSurfStd", Psurf)

; read optional fields common to both standard and support products

if arg_present(PSurf) then ret = EOS_SW_READFIELD(SWid,"PSurfStd", Psurf)
if arg_present(TSurf) then ret = EOS_SW_READFIELD(SWid,"TSurfStd", TSurf)
if arg_present(TSurf_Air) then ret = EOS_SW_READFIELD(SWid,"TSurfAir", tsurf_air)
if arg_present(LandFrac) then ret = EOS_SW_READFIELD(SWid,"landFrac", LandFrac)
if arg_present(sun_glint_distance) then ret = EOS_SW_READFIELD(SWid,"sun_glint_distance",
sun_glint_distance)
if arg_present(final_clear_flag) then ret = EOS_SW_READFIELD(SWid,"clear_flag", clear_flag)
if arg_present(Invalid) then ret = EOS_SW_READFIELD(SWid,"Invalid", Invalid)

if standard eq 1 then begin          ; reading standard product

; read required fields which are different in standard product

ret = EOS_SW_READFIELD(SWid,"pressStd", pres)
ret = EOS_SW_READFIELD(SWid,"TAirStd", tair)
ret = EOS_SW_READFIELD(SWid,"H2OMMRStd", h2o)
ret = EOS_SW_READFIELD(SWid,"O3VMRStd", ozo)
ret = EOS_SW_READFIELD(SWid,"nSurfStd", NSurf)
sz = size(pres)
numpres = sz[1]

; read optional fields (keyworded) which are in standard product

if arg_present(CldFrc) then ret = EOS_SW_READFIELD(SWid,"CldFrcStd", CldFrc)
if arg_present(TCldTop) then ret = EOS_SW_READFIELD(SWid,"TCldTopStd", TCldTop)
if arg_present(PCldTop) then ret = EOS_SW_READFIELD(SWid,"PCldTopStd", PCldTop)
if arg_present(clear_flag_4um) then ret = EOS_SW_READFIELD(SWid,"clear_flag_4um", clear_flag_4um)
if arg_present(clear_flag_11um) then ret = EOS_SW_READFIELD(SWid,"clear_flag_11um",
clear_flag_11um)

endif else begin                    ; reading support product

; read required fields which are different in support product

ret = EOS_SW_READFIELD(SWid,"pressSupp", pres)
ret = EOS_SW_READFIELD(SWid,"TAirSup", tair)
ret = EOS_SW_READFIELD(SWid,"H2OCDSup", h2o)
ret = EOS_SW_READFIELD(SWid,"O3CDSup", ozo)
ret = EOS_SW_READFIELD(SWid,"nSurfSup", NSurf)
sz = size(pres)
numpres = sz[1]

```

```
; read optional fields (keyworded) which are in support product

  if arg_present(ts_forecast) then ret = EOS_SW_READFIELD(SWid,"tsurf_forecast", ts_forecast)
  if arg_present(tsurf_diff_1lum) then ret = EOS_SW_READFIELD(SWid,"tsurf_diff_1lum",
tsurf_diff_1lum)
  if arg_present(spatial_coh_1lum) then ret = EOS_SW_READFIELD(SWid,"spatial_coh_1lum",
spatial_coh_1lum)
  if arg_present(tsurf_diff_4um) then ret = EOS_SW_READFIELD(SWid,"tsurf_diff_4um", tsurf_diff_4um)
  if arg_present(spatial_coh_4um) then ret = EOS_SW_READFIELD(SWid,"spatial_coh_4um",
spatial_coh_4um)
  if arg_present(cldfrcvis) then ret = EOS_SW_READFIELD(SWid,"CldFrcVis", cldfrcvis)

endelse          ; finished reading support product

print, "done reading - detaching and closing the file"
ret = EOS_SW_DETACH(SWid)
ret = EOS_SW_CLOSE(fid)

end
```